



## РЕФЕРАТ

Выпускная квалификационная работа по теме "Разработка системы "Информатор" содержит 47 страниц текстового документа, 1 приложение, 15 использованных источников, 25 иллюстраций, 7 таблиц.

ВЕБ–ПРИЛОЖЕНИЕ, БАЗА ДАННЫХ, C#, СУБД, HTML, MS SQL, ASP.NET, ЗАПРОС SQL, ВЕБ–СЕРВЕР, ИНФОРМАЦИЯ

Объектом исследования является веб–технология ASP.NET.

Предмет исследования: применение технологий ASP.NET, ADO.NET для создания информационной системы "Информатор" на базе веб–приложения.

Цель работы.

Разработать информационную систему "Информатор" на базе веб–приложения, которая будет информировать пользователей о некоторых запланированных событиях.

Задачи:

- разработать структуру базы данных, которая будет использоваться в веб-приложении;
- наполнить базу данных тестовой информацией;
- ознакомиться с технологией ASP.NET для разработки веб–приложения;
- разработать веб–приложение, взаимодействующее с базой данных, используя технологию ASP.NET.

Актуальность создания информационной системы любого вида – донесение информации до широкого круга пользователей этой системы. Скорость обмена информацией в современном мире постоянно растет, поэтому необходимо иметь средства позволяющие обеспечивать этот обмен.

Реализована система, позволяющая пользователям отслеживать изменения, происходящие в учебном процессе, а так же вести переписку с активными пользователями системы.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Теоретическая часть.....	5
1.1 Языки программирования.....	5
1.2 Среды разработки веб–приложений .....	10
1.3 Аутентификация и разграничение прав доступа.....	16
2 База данных.....	20
2.1 Нормализация базы данных .....	20
2.2 Модель предметной области .....	23
2.3 Инфологическая модель .....	25
2.4 Создание базы данных .....	27
3 Создание веб–приложения "Информатор" .....	31
3.1 Начало работы в Visual Studio.....	31
3.2 Основные элементы создания веб–приложения .....	35
3.3 Дизайн пользовательского интерфейса.....	37
3.4 Реализация веб–приложения .....	38
ЗАКЛЮЧЕНИЕ .....	45
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	46
ПРИЛОЖЕНИЕ А (обязательное) Сущности и их атрибуты.....	48

## ВВЕДЕНИЕ

С развитием технологий и средств связи кардинально меняется и образ жизни человека. Современный человек вынужден использовать различные технологии для общения, причем не только повседневного, но и делового. Не осталось в стороне от этих изменений и образование. Теперь, для осуществления учебного процесса уже не достаточно личного контакта преподавателя и студента. В школах повсеместно вводятся "электронные журналы", в университетах – "электронные курсы".

На данный момент существует необходимость общения студентов с преподавателем для коррекции процесса обучения, обсуждения учебных вопросов и обмена информацией.

В данной работе создадим информационную систему "Информатор" в виде веб–приложения, для обмена информацией между группами студентов и преподавателями. В основе информационной системы лежит база данных, где храниться основная информация. Эта информация хранится и обрабатывается в результате запросов. Наша цель создать базу данных и интуитивно–понятный интерфейс для работы с ней. Для создания и развития веб–приложения потребуется понимать и использовать различные технологии. Разработка веб–приложения включает три основных компонента: веб–сервер, язык программирования запросов, исполняемых на стороне сервера, и базу данных.

Для дальнейшей организации работы, решим следующие задачи:

- изучить специальное программное обеспечение для создания реляционных баз данных MS SQL Server;
- определиться со структурой баз данных;
- освоить язык программирования, на котором будет написан сайт (C# и язык гипертекстовой разметки HTML);
- создать веб–приложение на основе технологии ASP.NET.

## **1 Теоретическая часть**

Разработка информационной системы практически всегда означает, что будет происходить работа с базой данных. Пользователей, взаимодействующих с базой данных, будет несколько групп: студенты, преподаватели и администраторы. Поэтому для исключения конфликтов необходима функция учета различных запросов. Для этого нужно ввести долгосрочную функцию управления, направленную на регулирование и выполнение всех этапов проектирования. За сохранность данных отвечают администраторы.

В данном разделе рассмотрим: несколько широко распространенных языков программирования; формирование базы данных и ее структуры; обеспечение безопасности с использованием разграничения прав доступа.

Термин "Информация" многогранен. Происходит от латинского слова "Informatio" – сведения, разъяснения, изложение. Определим понятие информация как – какие-либо необходимые для хранения и передачи данные.

### **1.1 Языки программирования**

В основе любого программирования лежит создание и использование алгоритмов. Написание алгоритмов реализуется за счет языков программирования. Разработка приложений происходит при помощи написания кода на специализированном языке программирования.

Существует множество языков программирования: C/C++, C#, Perl, Python, JavaScript, Ruby, PHP, Visual Basic, Java и т.д. Истоками большинства современных языков являются такие языки программирования как ассемблер, Fortran, Pascal и т.д. (Рисунок 1).

В основе всех языков лежит машинный код – машинный язык, система команд определенной вычислительной машины, которая интерпретируется процессором или микропрограммой этой машины.

В реальности электронно–вычислительные машины (ЭВМ) работают по принципу нулей и единиц: в зависимости от перепада напряжения формируется последовательность нулей и единиц, что создает машинный код.

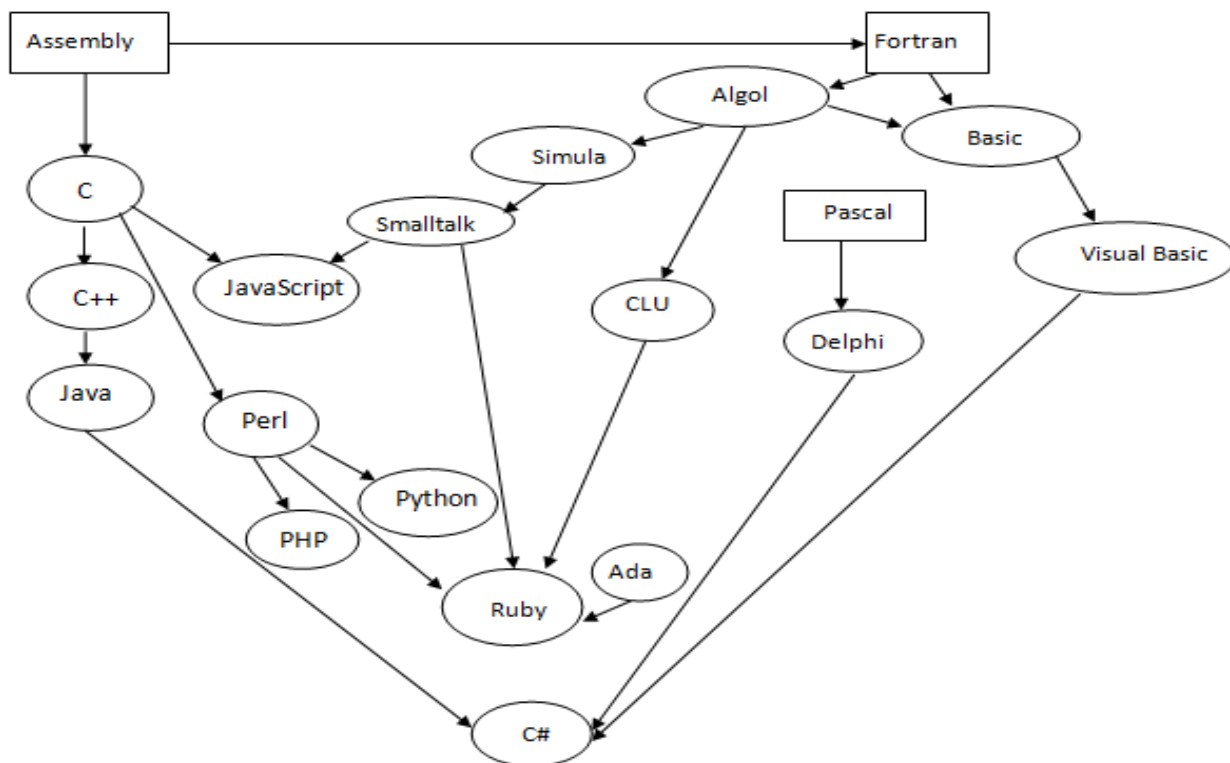


Рисунок 1 – История развития языков программирования

Для разработки веб–приложения выделяют две группы: клиентские и серверные языки программирования.

Программы на клиентских языках обрабатываются на стороне пользователя и выполняются браузером.

Серверный язык программирования выполняет программу, связанную с сайтом или приложением. Вызванный сайт обрабатывается на сервере, реализуя все программы своей страницы, и возвращается к пользователю в виде файла.

Рассмотрим несколько популярных языков веб–программирования на рисунке 2.

## Perl

Расшифровывается как Practical Extraction and Report Language – практический язык для извлечения данных и составления отчетов.

Язык программирования общего назначения, разработанный для манипуляций с текстом. Применяется для выполнения: системного администрирования, веб-разработок, сетевого программирования, игр, биоинформатики, разработку графических пользовательских интерфейсов.

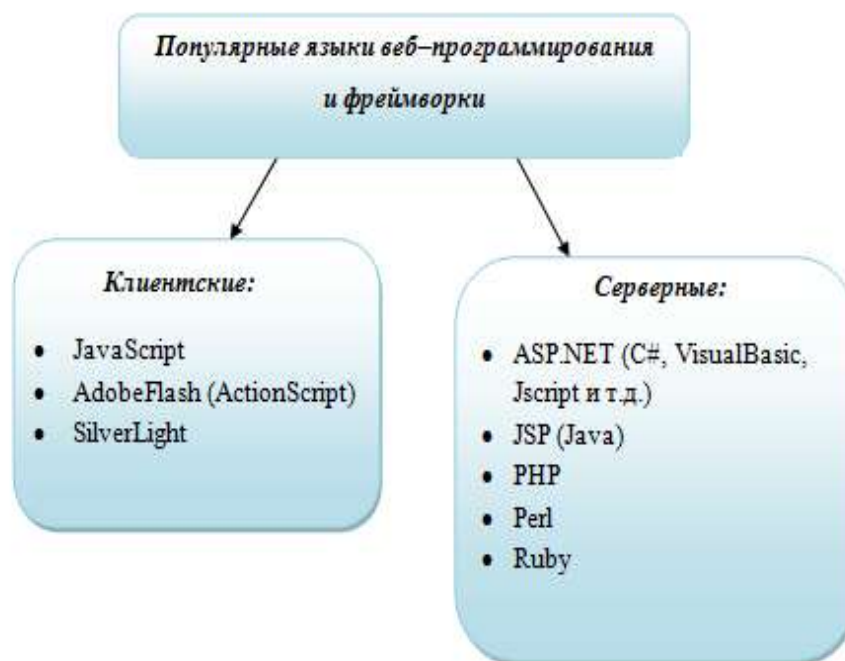


Рисунок 2 – Языки веб-программирования

Практичный язык, легкий в использовании, эффективный, но не минималистичный. Главное достоинство – поддержка различных стилей программирования, контроль за памятью, встроенная поддержка обработки текста и большая коллекция модулей сторонних разработчиков.

Структура Perl ведет свое начало от языка C.

### **Python**

Высокоуровневый язык программирования общего назначения. Простой и выразительный синтаксис, отсутствие скрытых переменных. Стандартная библиотека включает большой объем полезных функций. К преимуществам так же относятся: динамическая типизация (тип переменной определяется только во время применения), автоматическое управление памятью, полный

самоанализ, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных.

Чаще всего его сравнивают с Perl за успешное применение в написании скриптов, и с Ruby как с разумно разработанной системой для объектно-ориентированного программирования.

### **JavaScript**

JavaScript – быстрый, мощный и безопасный объектно-ориентированный язык программирования общего назначения. Создан для реализации динамических веб-страниц в браузерах.

Основные архитектурные черты: динамическая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

Язык JavaScript не принадлежит какой-либо компании или организации.

### **Ruby**

Язык программирования, объединивший в себе лучшее от таких языков как Perl, Python, Smalltalk, Lisp, Eiffel, без их недостатков.

Динамический, рефлексивный, интерпретируемый высокоуровневый язык программирования. Обладает динамической типизацией, многопоточностью, автоматической сборкой мусора.

По синтаксису ближе к Perl. Недостатком является непростое обучение и недостаточное количество информации для изучения данного языка, а так же его медленное развитие.

### **PHP**

Personal Home Pages, что переводится как личные домашние страницы, или инструмент для создания персональных веб-страниц. Рекурсивный акроним – Hypertext Preprocessor – препроцессор гипертекста.

Распространенный язык программирования общего назначения с открытым исходным кодом. Сконструирован специально для ведения веб-разработок и его код может внедряться непосредственно в HTML, а так же для



работ с базами данных. Файл с расширением php содержит комбинацию HTML–тегов и скриптов, запускаемых для выполнения на веб–сервере.

Отличие от JavaScript: php–скрипты выполняются на сервере и генерируют HTML, который посылается пользователю.

Прост в освоении, многофункционален. Поддерживается большинством хостинг–провайдеров. Обладает динамической типизацией, интерпретатором. Взаимодействует с большим количеством систем управления базами данных.

Синтаксис PHP схож с синтаксисом языка C. Некоторые элементы, такие как ассоциативные массивы и цикл foreach, заимствованы из Perl.

Для работы программы не требуется описывать какие–либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP. Из недостатков стоит упомянуть о не безопасности веб–приложений, написанных на данном языке.

## **HTML**

Это язык гипертекстовой разметки. Отвечает за расположение текста, рисунков, таблиц и любой другой информации в документе, или иными словами – HTML отвечает за оформление документа. Основные команды называются дескрипторами или тегами, большинство из которых парные, то есть состоят из открывающего и закрывающего тегов.

## **CSS**

Cascading Style Sheets – каскадные таблицы стилей – формальный язык, описывающий внешний вид документа, написанного при помощи языка разметки. Используется для оформления веб–сайта.

## **C Sharp**

Объектно–ориентированный язык программирования. C# относится к семье языков с синтаксисом от C (наиболее близкие из них C++ и Java). Имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, атрибуты, обработку событий и исключений, указатели на функции–членов классов, использование веб–технологий (XML, SOAP).

К недостаткам можно отнести:

- сложный синтаксис;
- не высокая производительность;
- не кросс–платформенный.

### **Структурированный язык запросов SQL**

Structured Query Language – язык, с помощью которого возможна работа с данными в реляционных базах данных. Для обращения к базе данных используются запросы, написанные на языке SQL. Запрос – это команда, которая передается серверу базы данных, и сообщает ему, какую информацию необходимо вывести из таблицы в память. Эта информация посылается на экран компьютера. Прост в освоении.

Язык SQL подразделяется на четыре группы команд:

- язык определения данных DDL – используется для создания, изменения и удаления всевозможных объектов базы данных (CREATE, ALTER, DROP);
- язык манипулирования данными DML – осуществляет возможность перемещения данных в базе, их извлечения, изменения и удаления из нее (INSERT, UPDATE, DELETE);
- язык управления доступом к данным DCL – применяется для управления правами доступа к определенным объектам базы данных (GRANT, REVOKE, SET ROLE);
- язык управления транзакциями TCL – позволяет контролировать обработку транзакций в базе данных (COMMIT, ROLLBACK, SAVEPOINT).

## **1.2 Среды разработки веб–приложений**

Неотъемлемой частью при разработке веб–приложений является выбор веб–сервера и исполняющей среды. Рассмотрим некоторые технологии для создания веб–сайта, и остановимся на наиболее оптимальном варианте.

Веб – web – сеть, паутина, интернет–пространство.

При использовании этой приставки подразумевается, что слово с ним употребляемое относится, в той или иной степени, к интернет (веб–приложение, веб–сайт, веб–сервер, веб–камера и т.д.).

### **Веб–сервер**

Сервер – это программный элемент вычислительной системы, исполняющий обслуживающие функции по запросу клиента, и разрешая ему доступ к определенным ресурсам или услугам.

Веб–сервер – сервер, принимающий HTTP–запросы от клиентов и предоставляющий им HTTP–ответы с HTML–страницей, изображением, документом, медиа–поток или другими данными.

Веб–сервер является программным обеспечением, которое выполняет функции веб–сервера, и компьютером, на котором работает это программное обеспечение. Таким образом, выполняется следующий алгоритм действий:

- пользователь вводит свой вопрос или ссылку в браузер;
- браузер делает запрос при помощи URL (определитель местоположения ресурса);
- веб–сервер принимает и обрабатывает запрос, и находит соответствующий запросу физический файл в системе URL;
- веб–сервер определяет наличие прав доступа к данному файлу и возвращает его браузеру;
- ведет запись протокола, с помощью которого составляется статистика посещаемости какого–либо сайта.

HTTP – Hyper Text Transfer Protocol – протокол передачи гипертекста. В основе HTTP находится технология "клиент–сервер", следовательно: существуют потребители (клиенты), которые посылают запрос, и поставщики (серверы), которые обрабатывают этот запрос и возвращают ответ.

### **Веб–приложение**

Так как не существует конкретного определения, и граница между понятиями сайта и приложения размыта, предположим, что веб–приложение это интерактивный веб–сайт. В качестве клиента выступает браузер, а сервера –

веб-сервер. Данные хранятся на сервере, а обмен информацией осуществляется по сети. Веб-приложения являются кроссплатформенными сервисами, потому что клиенты не зависят от конкретной операционной системы пользователя.

На рисунке 3 показано как происходит запрос от браузера к веб-приложению, которое использует базу данных для хранения данных, и как оно возвращает ответ на запрос.



Рисунок 3 – Взаимодействие веб-приложения с веб-клиентом

### Microsoft .NET Framework

Программная платформа, предназначенная для совместимости разных служб, написанных на различных языках, с помощью общеязыковой среды исполнения CLR (Common Language Runtime – один из основных компонентов пакета Microsoft .NET Framework). Помогает устранять конфликты между разными версиями сборок.

В состав платформы .NET Framework входят:

- среда CLR и библиотеки;
- ADO.NET;
- ASP.NET;
- Windows Forms;
- веб-службы;
- компиляторы.

Языки программирования в .NET: C#, Visual Basic.NET, JScript.NET, C++/CLI, F#, J#.

ADO.NET – (ActiveX Date Objects – объекты данных ActiveX) технология состоящая из набора классов, предоставляющих доступ к данным. С помощью ADO.NET устанавливаются подключения, осуществляется отправка запросов к базам данных, получение ответа от базы данных и производится ряд других операций.

ASP.NET – это веб–платформа, предоставляющая все необходимые службы для создания серверных веб–приложений. ASP.NET создана на основе платформы .NET Framework, поэтому все функции .NET Framework доступны для приложений ASP.NET. Приложения могут быть написаны на любом языке, совместимом со средой CLR, включая Visual Basic и C#.

Для создания веб–приложений ASP.NET можно воспользоваться Visual Studio. Помимо этого, имеется бесплатный самостоятельный продукт Visual Studio Express для Web, который включает в себя базовый набор функций веб–разработки, реализованный в Visual Studio.

### **Microsoft Visual Studio**

Microsoft Visual Studio (в дальнейшем VS) является комплексной средой разработки, которая может использоваться для создания мощных и надежных корпоративных веб–решений. VS представляет разные языки программирования, которые доступны в процессе разработки приложений .Net Framework. В нем осуществляется возможность написания кода и создания компонентов с помощью двух основанных языков .NET Framework: Microsoft Visual C# и Microsoft Visual Basic.

В Visual Studio реализуется разработка веб–приложений с помощью масштабируемых, многоразовых серверных компонентов, которые предоставляют возможность повысить производительность и помогают результативно создавать веб–приложения и веб–сайты ASP.NET.

### **Веб–сервер Microsoft IIS**

Internet Information Server – комплекс веб–служб от Microsoft. С помощью

данного сервера разрабатываются сайты с использованием технологии ASP.NET (существует возможность разработки сайтов и на других языках: PHP, Perl и т.д.).

Основные компоненты IIS, позволяющие управлять веб-сайтом:

- веб-службы;
- служба FTP (протокол передачи файлов);
- SMTP сервер (протокол передачи почты).

Обладает графическим интерфейсом для удобства установки и управления.

IIS состоит из нескольких модулей, которые выполняют важные функции для приложений и веб-ролей сервера. У каждого модуля свои обязанности. Модули делятся на нативные (Native) и управляемые (Managed). Нативные модули поставляются вместе с IIS, а управляемые – в составе .NET Framework (ASP.NET). Взаимодействовать из кода своего ASP.NET-сайта возможно только с управляемыми модулями.

Драйвер HTTP.sys – прослушивает запросы к серверу (протоколы).

Служба публикации (World Wide Web Publishing) WWW (W3SVC) – адаптер, отдельный модуль для работы с протоколами HTTP и управления рабочими процессами приложений, выполняющий функции: администрирования драйвера HTTP.sys, управления процессом, мониторинг показателей производительности веб-сайтов (ведется счетчик производительности).

Служба активации WAS (Windows Process Activation Service) – выполняет управление рабочими процессами. При переходе запроса к HTTP.sys, происходит обращение к службе WAS для получения рабочего процесса приложения, которому передается запрос для обработки и формирования ответа клиенту.

Обработка запроса и вывод ответа происходит в IIS по следующей схеме:

- браузер обращается к веб-серверу по конкретно-заданному URL, на стороне сервера запрос прослушивает драйвер HTTP.sys;

- HTTP.sys обращается к WAS для получения информации из хранилища конфигурации;
- служба WAS запрашивает конфигурацию из хранилища – из файла в папке IIS (applicationHost.config);
- так как, данный запрос получен по протоколу HTTP, конфигурационную информацию получает служба W3SVC (WWW Service), эта информация содержит в себе данные о пуле приложений (application pool) и прочих параметрах сайта;
- служба W3SVC использует эту информацию для конфигурации HTTP.sys;
- служба WAS запускает процесс W3WP.exe для пула приложений, если он ещё не был запущен;
- в процессе W3WP.exe работает приложение веб-сайта, которое, формирует и возвращает ответ драйверу HTTP.sys;
- HTTP.sys отправляет ответ браузеру.

Одна из популярных платформ наряду с Apache.

### **Средство создания веб-сценариев PHP и веб-сервер Apache**

Для узлов, которым нет необходимости в крупномасштабном, мощном Web-приложении, сервер Apache примечателен тем, что вместе с PHP образует эффективную комбинацию средств, отличающуюся высокой безопасностью.

Сервер Apache характеризуется универсальной поддержкой различных платформ, оснащен обширной документацией, благодаря чему весьма популярен в сообществе Интернет.

Основными преимуществами Apache считаются надёжность и гибкость конфигурации. С помощью него осуществляется подключение внешних модулей для передачи данных, использование системы управления базами данных (далее СУБД) для аутентификации пользователей, модифицирование сообщений об ошибках и т. д.

Для определения подходящего веб-сервера применим таблицу 1.

Таблица 1 – Сравнительные характеристики IIS и Apache

Функция	Apache	IIS
Поддерживаемая ОС	Windows, Unix, Linux, Mac OS	Windows
Техническая поддержка	сообщество	корпоративная
Разработка	свободное программное обеспечение (open-source)	проприетарное программное обеспечение (собственность Windows)
Безопасность	хорошо	отлично
Производительность	хорошо	хорошо
Стоимость	бесплатно	поставляется в комплекте с ОС Windows

### **Фреймворк** (Framework – каркас, структура)

Программная платформа, которая служит основой для будущего сайта. Обеспечивает высокоуровневую или общую функциональность приложения, предназначенную для ускорения разработки приложений. Создание сайта на фреймворке происходит с нуля, на всех стадиях разработки.

Создается:

- публичная часть;
- база данных;
- алгоритмы для частей системы;
- администрирование;
- интерфейс.

На разработку сайта с фреймворк уходит большое количество времени, а результат получается индивидуальным.

Подведем итоги: для разработки веб-приложения выберем среду разработки Microsoft Visual Studio, веб-сервер от Microsoft IIS, технологию для разработки веб-проектов ASP.NET в составе VS, приложение будет написано на языке C#, а веб-страницы будут оформлены с помощью HTML кода и CSS.

## **1.3 Аутентификация и разграничение прав доступа**

Необходимым условием работы любой информационной системы является безопасность. Защита производится от случайного или умышленного



вмешательства в деятельность ИС, от несанкционированного доступа и хищения информации, от намеренного внесения изменений, приводящих систему в негодность и т.д. Для рассмотрения способов обеспечения безопасностью информационной системы введем несколько основных понятий.

Идентификация – процедура распознавания пользователя по его логину, или установления себя в системе как определенного пользователя. Характеристикой пользователя является идентификатор.

Аутентификация – процедура проверки подлинности идентификатора пользователя. Установление соответствия идентификатора в данной системе.

Авторизация – процедура предоставления пользователю определенных прав или проверка наличия прав, при выполнении какого-либо действия в системе.

Для адекватной и полной работы в системе вводят разграничение прав доступа. То есть каждую группу пользователей наделяют определенными правами, разрешая или запрещая какие либо действия с данными (например: удалять, сохранять, записывать, читать и т.д.). Такая система является главной защитой от несанкционированного доступа к информации или порче самой системы.

Отметим существование двух принципов разграничения по управлению доступом: дискреционное и мандатное.

Дискреционное управление доступом основывается на применении списков контроля доступа, назначаемых каждому субъекту системы. Субъекту обладающему правом доступа к определенному объекту, будет отказано в доступе, при попытке доступа к другому объекту. Субъект с определенным правом доступа может передать это право любому другому субъекту. Недостаток: при внесении в систему каких-либо изменений, связанных с пользователями, возникает необходимость пересматривать права доступа каждого субъекта к каждому объекту.

Мандатное управление доступом реализуется за счет сопоставления каждому субъекту и объекту классификационных меток, которые отображают

место данного субъекта (объекта) соответственно иерархии. С помощью этих меток субъектам и объектам должны быть назначены классификационные уровни, которые являются комбинациями иерархической и неиерархической категории. Такие метки служат основой мандатного принципа разграничения доступа.

Выделяют четыре метода разграничения доступа приведенных в таблице 2.

Таблица 2 – Методы разграничения доступа

Методы разграничения доступа	Основные особенности
По спискам	<ul style="list-style-type: none"> <li>– каждому пользователю – список ресурсов и прав доступа к ним;</li> <li>– каждому ресурсу – список пользователей и их прав доступа к данному ресурсу.</li> </ul> <p>Позволяют установить права с точностью до пользователя. Добавлять или запрещать права доступа возможно без труда.</p>
Парольное	<p>разграничение прав пользователя происходит по паролю. Используются все методы парольной защиты. Менее удобен.</p>
По уровню секретности и категориям	<ul style="list-style-type: none"> <li>– по уровням секретности; имеет подуровни: общий доступ, конфиденциально, секретно, совершенно секретно (приоритет пользователей задается в соответствии с уровнем секретности, чем выше приоритет – тем больше полномочий);</li> </ul>
	<ul style="list-style-type: none"> <li>– на категории. Все ресурсы системы классифицируют и разделяют по уровню важности. Определенному уровню соответствует своя группа пользователей.</li> </ul>
Матрица установления полномочий	<ul style="list-style-type: none"> <li>– все права доступа, полномочия, пользователи, ресурсы, ссылки являются элементами матрицы доступа. Унифицированный и удобный подход, но слишком емкостный, и не совсем оптимален в использовании ресурсов (много пустых клеток).</li> </ul>

Чаще всего, на практике сочетают различные методы разграничения доступа. Например, первый, третий и четвертый методы усиливают парольной защитой.

Из этого можно сделать вывод, что методы разграничения доступа по уровням секретности и категориям представляют собой примеры мандатного управления доступом.

Для организации входа в систему будем придерживаться следующей схемы, показанной на рисунке 4.

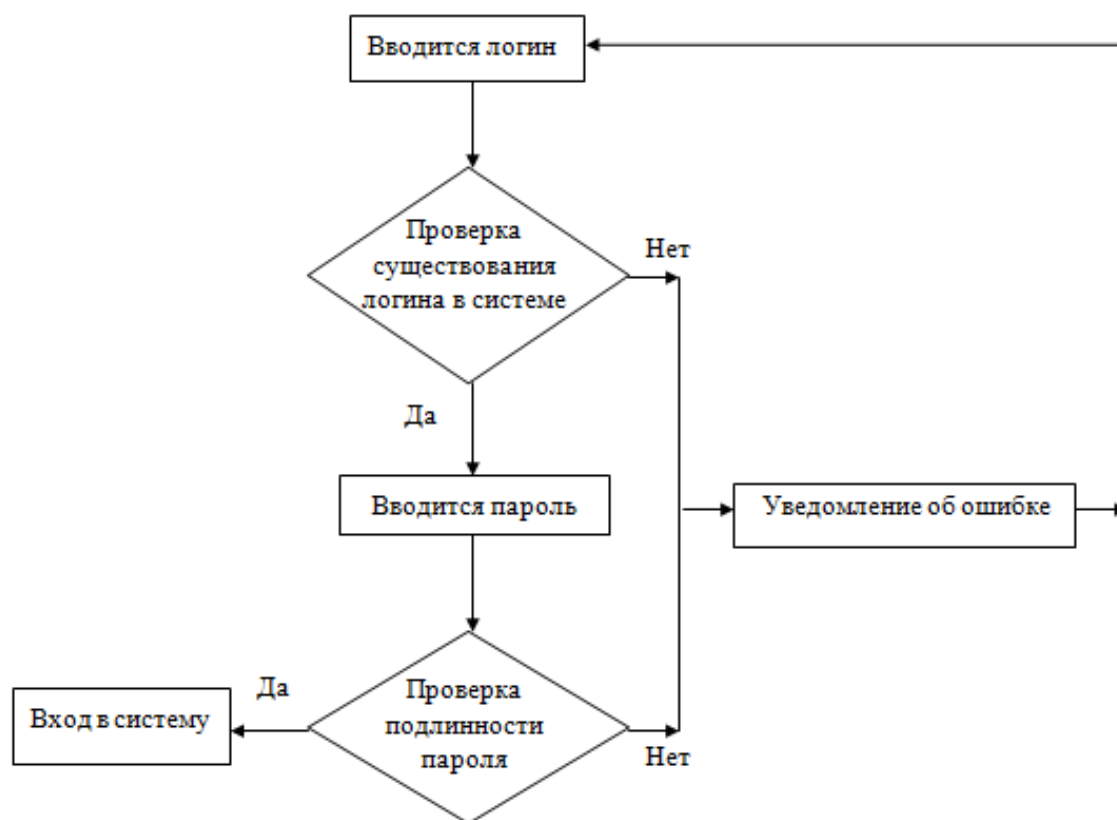


Рисунок 4 – Алгоритм доступа

## **2 База данных**

База данных – это совокупность данных, которая хранится в определенном порядке. Система управления базами данных – набор некоторых языковых и программных средств, за счет которых реализуется доступ к данным, управление ими, возможность создавать, изменять, удалять данные, а так же обеспечивается безопасность хранимых данных, и много другое.

База данных должна быть непротиворечивой, простой и целостной, корректной, не должна содержать лишней информации и обеспечивать безопасность вводимых данных, легко изменяемой и актуальной, а так же, удовлетворять требованиям пользователей. Для создания базы данных оптимальным языком управления является структурированный язык запросов SQL.

### **2.1 Нормализация базы данных**

Нормализация – это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

Большое количество не нужной информации приводит к расходованию свободного места на диске, затрудняет обслуживание базы данных и работу оборудования. К тому же повторение данных в разных местах, при их изменении, требует изменять их во всех местах.

Процесс нормализации определяется шестью нормальными формами, но на практике достаточно привести базу данных к третьей нормальной форме.

Каждой нормальной форме соответствует свой свод правил, при выполнении которого можно говорить о конкретной нормальной форме базы данных.

База данных представлена в виде связанных таблиц. Для приведения БД к нормальному виду необходимо рассматривать каждую таблицу отдельно.

При построении базы данных выделяют следующие объекты управления: сущность, атрибут, ключ, связь.

Сущность – объект, информация о котором хранится в базе данных. Примером могут быть: машины, люди, тарифы, животные, кафе, афиши и т.д.

Атрибут – качество, свойство, какая-либо характерная черта сущности. В общем, конкретная классифицированная информация о сущности. Например, для сущности машина атрибутами могут быть: цвет, марка модели, год выпуска, регистрационный номер и т.д.

Ключ – уникальный атрибут, по значению которого можно определить конкретного представителя сущности, обладающей этим ключом. По ключу сущность однозначно идентифицируется. Значение ключа не может быть пустым. В роли ключа может выступать набор значений атрибута, и такой ключ называется составным.

Связь – установленная логическая взаимосвязь между сущностями базы данных. За счет наделения сущностей первичными ключами, и с их помощью связывания таблиц происходит организация связи.

Существует три вида связи:

- один к одному – строка из первой таблицы сопоставляется с другой строкой из второй таблицы;
- один ко многим – строка из первой таблицы сопоставляется с несколькими строками из второй таблицы;
- много ко многим – строке из первой таблицы сопоставляется несколько строк из второй таблицы, и наоборот.

Таблица выступает в роли сущности, столбцы которой являются атрибутами, а строки – типы сущности, и ячейки – экземпляры сущности.

Рассмотрим три основные формы нормализации.

### **Первая нормальная форма**

Если в таблице на пересечении столбца со строкой имеется только одно

значение, а не группа или множество, то такое отношение будет находиться в первой нормальной форме. Выполняется свойство атомарности. Так же к правилам первой нормальной формы относится обязательное существование первичного ключа (или составного первичного ключа), а строки не должны зависеть друг от друга, то же относится и к столбцам.

Например, приведем к первой нормальной форме таблицу 3:

Таблица 3 – Студенты

№ Студенческого билета	№ Группы	ФИО	Сообщение
33551	3012	Степанов Степан Федорович	ID–ключ:123; Дата: 18.04.2012; Содержание: Добрый вечер... ID–ключ:121; Дата: 10.03.2012; №Содержание: Добрый день...

Из данной таблицы видно, что студент отправляет сообщения, которые в свою очередь содержат ряд других свойств (ID, дата, и содержание). К тому же столбец ФИО тоже не является однозначным и может быть разделен. В итоге мы разделили атрибут "Сообщение" на три столбца, по количеству его свойств. Таким образом, таблица 3 примет вид:

Таблица 4 – Студенты (первая нормальная форма)

№ Студенческого билета	№ Группы	ФИО	ID–ключ	Дата	Содержание
33551	3012	Степанов Степан Федорович	123	18.04.2012	Добрый вечер...
33551	3012	Степанов Степан Федорович	121	10.03.2012	Добрый день...

## Вторая нормальная форма

Теперь появляется необходимость избавиться от лишней информации. Для этого существует вторая нормальная форма. Стоит отметить что, всякая вторая нормальная форма, прежде всего, является первой, с дополнением в виде

ограничений. Ограничением будет являться следующее: любой не ключевой столбец, то есть атрибут, должен логически полно зависеть от первичного ключа.

Ключевым атрибутом в данной таблице является № Студенческого билета. Отсюда видно, что информация о сообщении должна быть вынесена в другую таблицу (таблица 5, 6).

Таблица 5 – Студенты (вторая нормальная форма)

№ Студенческого билета	№ Группы	Фамилия	Имя	Отчество
33551	3012	Степанов	Степан	Федорович

Таблица 6 – Сообщения (вторая нормальная форма)

ID–ключ	Дата	Содержание
123	18.04.2012	Добрый вечер...
121	10.03.2012	Добрый день...

### **Третья нормальная форма**

Если таблица находится во второй нормальной форме, и каждый не ключевой столбец нетранзитивно зависит от первичного ключа, то таблица находится в третьей нормальной форме.

Рассмотрим таблицу 5: у каждого студента свой идентификационный № студенческого билета. Каждый студент принадлежит какой-либо одной группе. При рассмотрении зависимостей между столбцами можно сделать вывод, что таблица уже находится в третьей нормальной форме.

## **2.2 Модель предметной области**

Предмет работы – приложение "Информатор". Клиенты приложения – студенты и преподаватели одного университета. Приложение осуществляет

общение между преподавателями и студентами. Сообщения хранятся в архиве приложения.

Сущности:

- студенты: № студенческого билета, № группы, фамилия, имя, отчество;
- группы :№ группы, курс, форма обучения;
- преподаватели: № преподавателя, фамилия, имя, отчество, № дисциплины;
- сообщения преподавателя: ID – ключ СП, дата/время, № преподавателя, содержание, № группы;
- сообщения студентов: ID – ключ СС, дата/время, № студенческого билета, текст сообщения, № преподавателя;
- почта: ID – ключ, ID – ключ СС, ID – ключ СП;
- предмет: № дисциплины, название предмета;
- расписание: ID – ключ, день недели, № дисциплины, № ленты, № группы;
- ленты: № ленты, время.

Семантические допущения:

- каждый студент может быть прикреплен только к одной группе;
- преподаватели могут быть прикреплены к нескольким группам студентов;
- преподаватели пишут сообщения в расписание своей группы;
- студенты могут писать в форуме своему преподавателю, комментировать сообщения преподавателя и друг друга;
- для каждой группы свое расписание.



## 2.3 Инфологическая модель

При построении инфологических моделей можно использовать язык ER–диаграмм (от англ. Entity–Relationship, т.е. сущность–связь), язык инфологического моделирования и языка "Таблицы – связи".

На основе представленных данных выделяем сущности:

- студенты;
- группы;
- преподаватели;
- сообщения преподавателя;
- предмет;
- расписание;
- ленты;
- сообщения студентов;
- почта.

Перечисляем атрибуты сущностей в таблице А.1 (Приложение А).

С учетом семантических допущений, полученную взаимосвязь между сущностями изображаем в виде ER–диаграммы на рисунке 5.

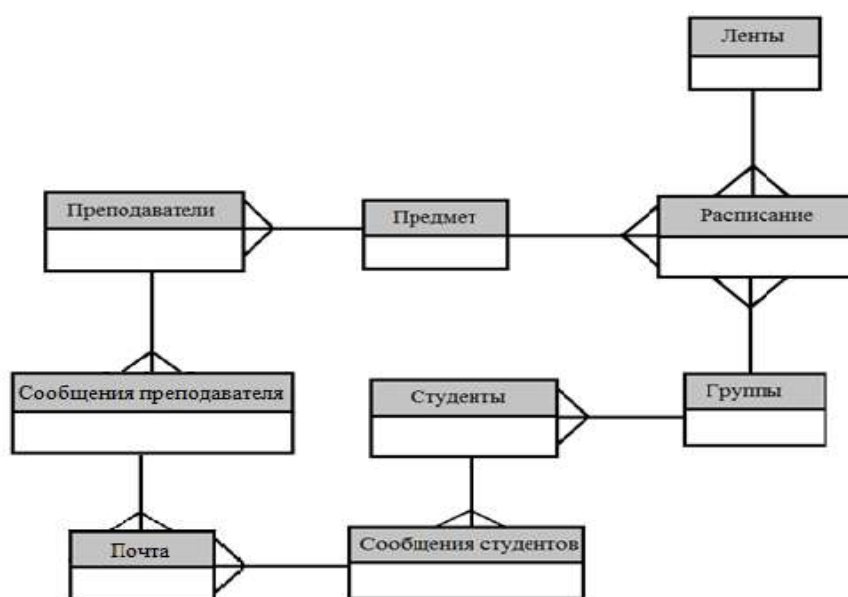


Рисунок 5 – ER–диаграмма

На основе ER–диаграммы создаем реляционную модель, преобразуя сущности в таблицы, а атрибуты сущности в столбцы соответствующих таблиц.

Создаем реляционную схему на рисунке 6, определяя первичные и внешние ключи, и связывая таблицы.

Выбор первичного ключа останавливаем на атрибуте сущности, у которого значение для всех экземпляров сущности различны, по которому сущность однозначно идентифицируется, и не может принимать пустых значений.

Проводим выбор внешних ключей, опираясь на построенную ER–диаграмму. Выбирая атрибуты ссылающейся сущности, обозначенной стрелкой "много", значения которых совпадают со значениями выбранного первичного ключа соседней сущности, участвующей во взаимосвязи.

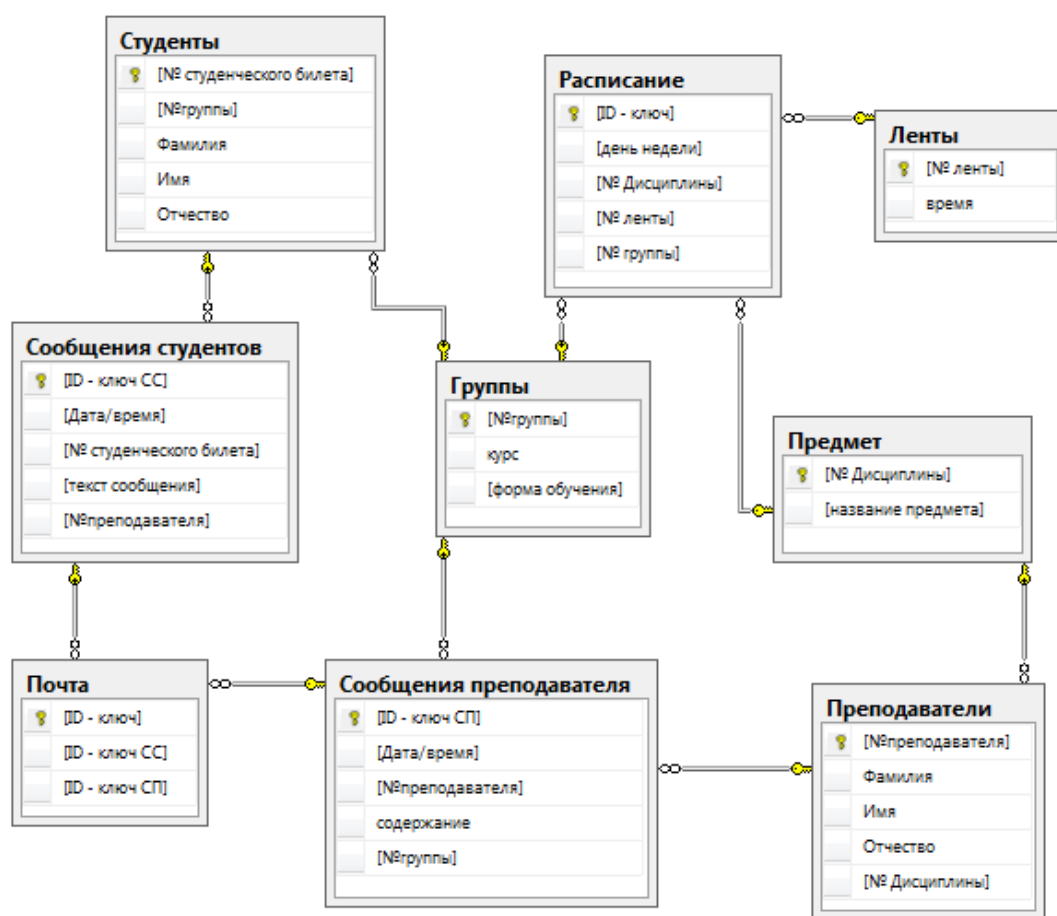


Рисунок 6 – Реляционная схема

## 2.4 Создание базы данных

База данных необходима нам для того, что бы сделать наше веб-приложение интерактивным, динамичным. Для ее создания была выбрана интегрированная среда SQL Server Management Studio.

Преимущества MS SQL Server:

- бесплатная;
- лицензионная;
- идеально подходит для независимых поставщиков программного обеспечения, пользователей серверов, веб-разработчиков и любителей;
- легко обновить до расширенных версий SQL Server;
- предусматривает поддержку баз данных размером до 4 Гб;
- доступная и простая программа от Microsoft.

Вся информация в базе данных имеет четкую структуру. Для вызова определенной информации используется язык структурированных запросов SQL. Обработка базы данных происходит через специальный SQL Server, который является специальным набором программ и утилит, с помощью которого создают и изменяют базы данных. MS SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft (далее СУБД).

В SQL Server Management Studio создаем базу данных. На рисунке 7 слева – показано окно, в котором задается имя базы данных, имена и пути к файлам базы данных, начальный размер файлов и шаг увеличения размера БД в случае необходимости, и справа – список баз данных, в котором отображена созданная БД.

Новая база данных – пуста, и не содержит таблиц. Поэтому следующим этапом будет создание таблиц.

Для создания таблиц используем команду create table:

```
create table Студенты ("№студенческого билета" int, "№группы" nvarchar(20), Фамилия nvarchar(20), Имя nvarchar(15), Отчество nvarchar(15));
```

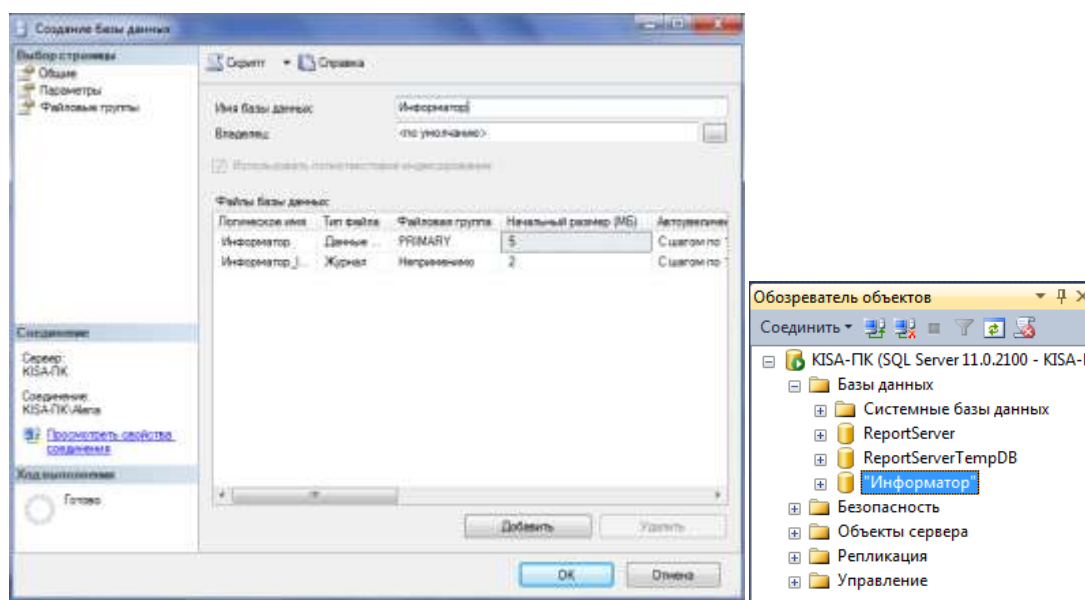


Рисунок 7 – Окно создания базы данных – слева, список баз данных – справа

Присваиваем каждому атрибуту имя, размер, тип данных.

Из поддерживаемых СУБД MS SQL типов данных мы будем использовать следующие:

- int – численные значения (целые) в диапазоне от  $(-2)^{31} - (-2\,147\,483\,648)$  до  $(2^{31} - 1) - (+2\,147\,483\,647)$ ;
- nvarchar – тип данных переменной длины в Юникоде для хранения текста до 4000 символов;
- nchar – символьный тип данных фиксированной длины;
- datetime – определяет дату и время в 24 – часовом формате включая секунды;
- nvarchar max – символьный тип данных переменной длины в Юникоде, где "max" указывает, что максимальный размер хранилища равен  $(2^{31} - 1)$  байт (2 ГБ).

При создании базы данных необходимо связать таблицы определенным образом. Делается это через задание таблицам первичных ключей, как показано на рисунке 8.

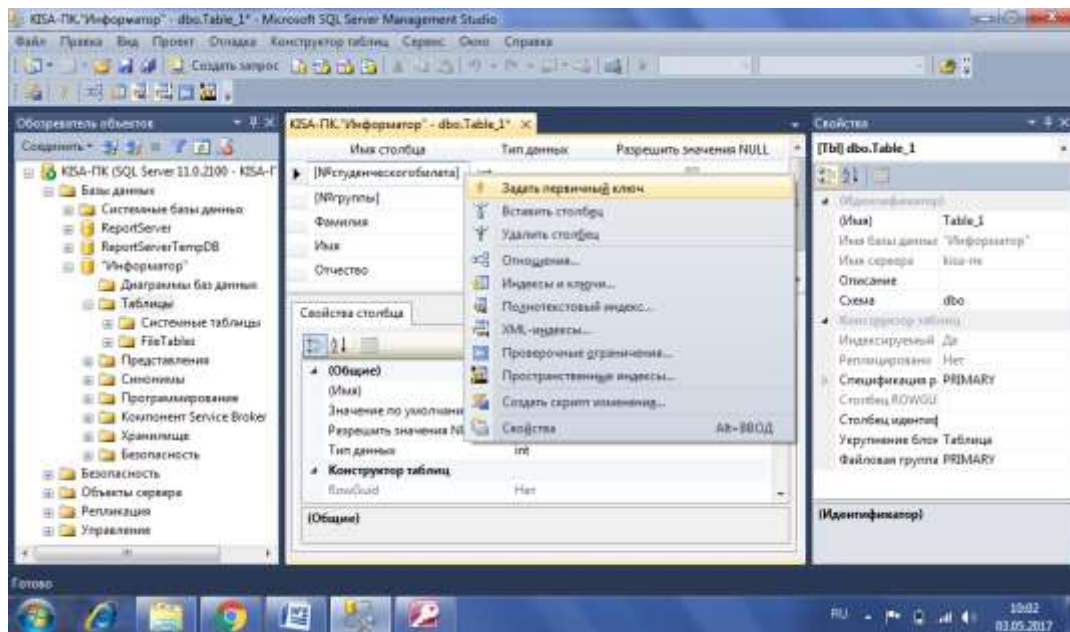


Рисунок 8 – Задание первичного ключа

Так же необходимо заполнить существующие таблицы данными. Для этого используем команду Insert:

INSERT INTO Студенты

([№студенческого билета],[№группы], Фамилия, Имя, Отчество)

VALUES ('111111','КИ12', 'Панкова', 'Анастасия', 'Николаевна');

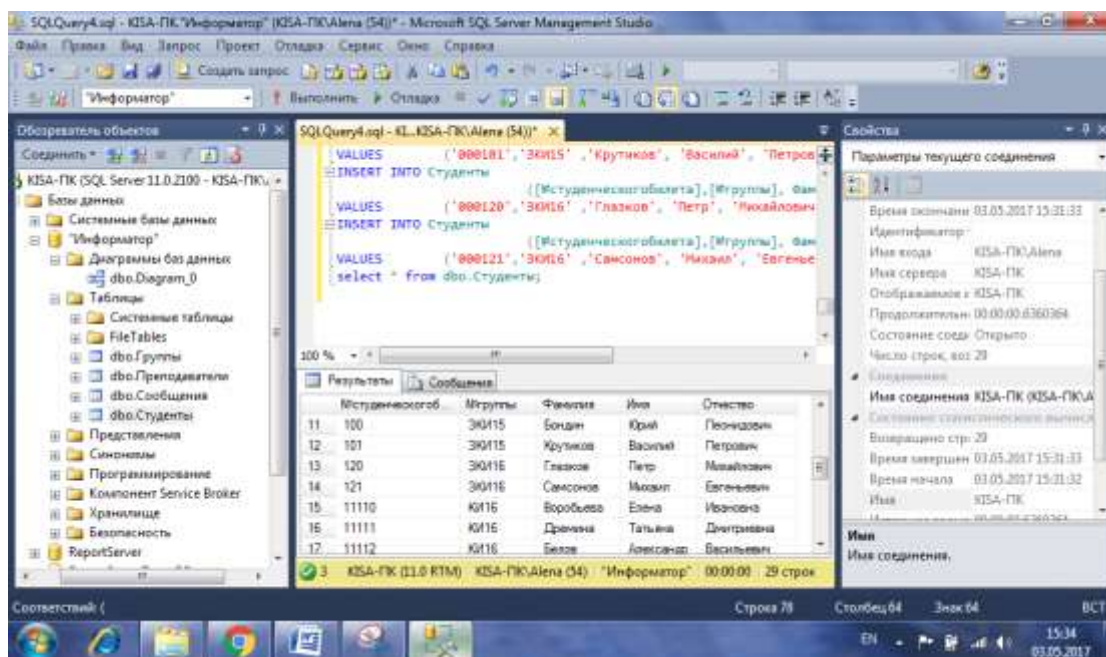


Рисунок 9 – Заполнение таблиц в среде Management Studio

Добавление пользователя БД.

Для предоставления доступа к БД необходимо создать учетную запись пользователя БД и связать с логином на уровне сервера БД:

```
CREATE USER user_name FOR LOGIN login_name
```

Для изменения свойств учетной записи: **ALTER USER.**

Для удаления учетной записи пользователя БД: **DROP USER.**

После создания и заполнения таблиц, создадим диаграмму базы данных. Для этого в обозревателе объектов в списке баз данных находим свою базу данных. Далее в своей базе данных кликаем правой кнопкой мыши по объекту "Диаграммы базы данных" и в выпадающем списке выбираем "создать диаграмму базы данных". Диаграмма отображает сущности и связи между ними. База данных готова.

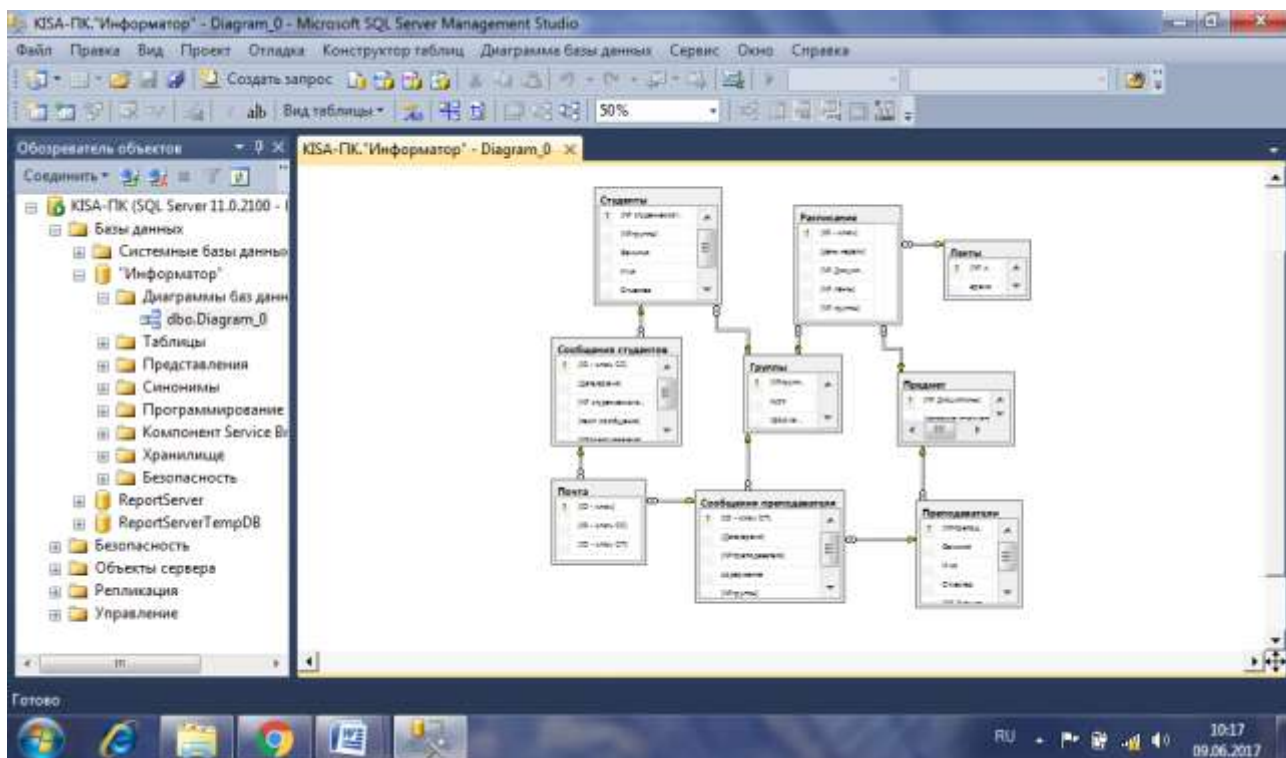


Рисунок 10 – Диаграмма базы данных



## 3 Создание веб–приложения "Информатор"

### 3.1 Начало работы в Visual Studio

Для создания веб–приложения была выбрана среда разработки MS Visual Studio Express для Web на языке C# (Рисунок 11).

Преимущества Visual Studio Express:

- бесплатная;
- лицензионная;
- простая в обучении и использовании программа от Microsoft;
- обладает редактором исходного кода, встроенным отладчиком машинного уровня и базовым набором функций веб–разработки;
- взаимодействует с MS SQL Server.

Является инструментом разработки .NET–приложений для Windows. В ней интегрированный дизайн форм, специализированный редактор кода, отладчик, мастер публикации и другие инструменты, необходимые программисту.

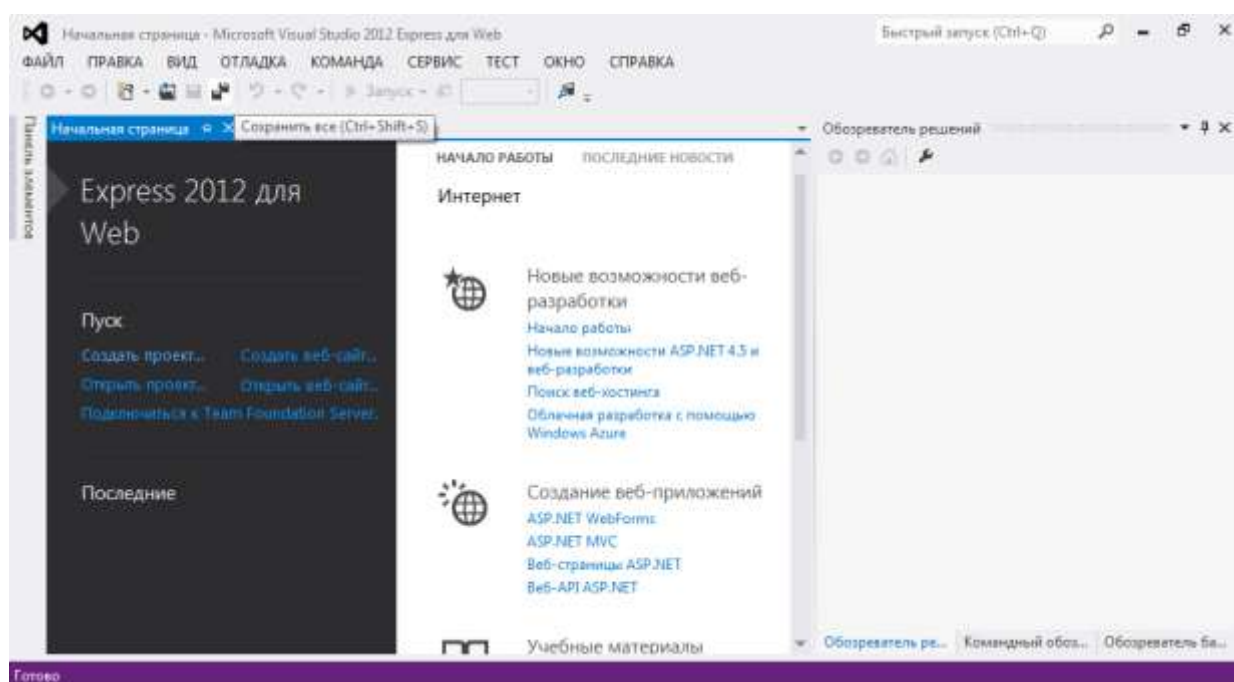


Рисунок 11 – Начальная страница Microsoft Visual Studio Express для Web

Создаем веб–приложение следующим образом:

- выбираем создать проект;
- Visual C#/ веб;
- приложение веб–форм ASP.NET Visual C#;
- в консольном меню выбираем "проект";
- добавить новый элемент, веб;
- Форма Web Form.

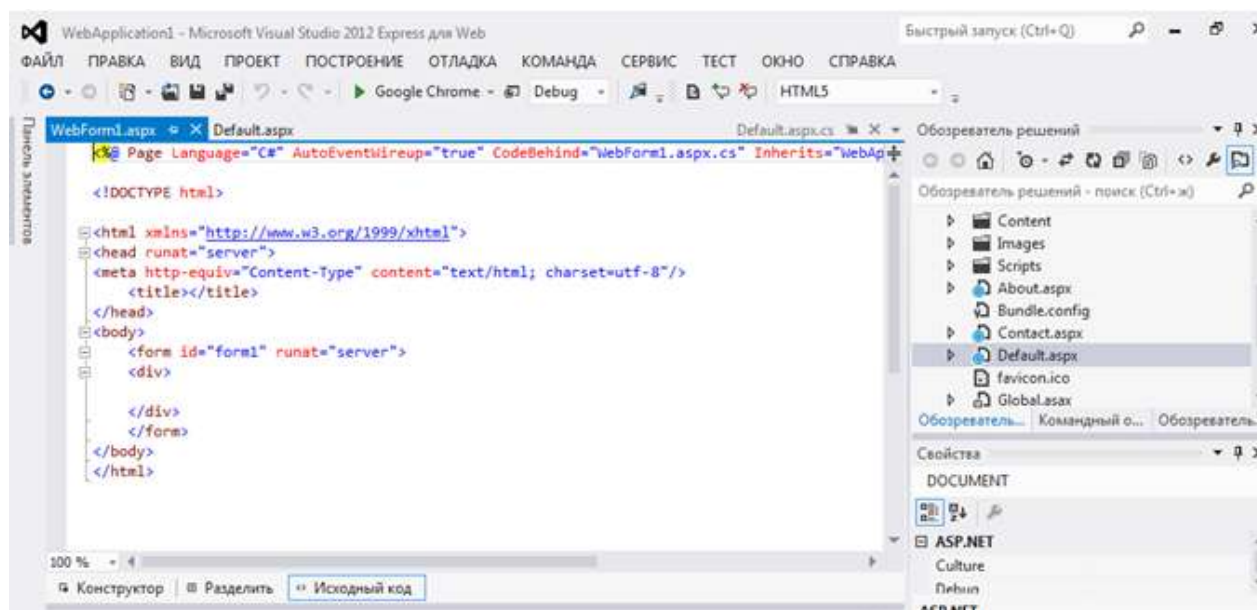


Рисунок 12 – Создание проекта WebApplication

Перед нами появляется окно с исходным кодом, который представляет собой расширенный HTML–файл. О необычности этого файла можно судить по элементу с дескрипторами `<%` и `%>` в первой строке, а так же атрибутам `runat` в элементах `head` и `form`.

Дескрипторы элементов управления в ASP.NET всегда начинаются с префикса `asp:`, за которым следует имя класса.

Осуществим привязку базы данных к приложению. Проект/ добавить класс/ данные/ Модель ADO.NET EDM (Рисунок 13).

Добавим стандартные элементы HTML для отладки интерфейса будущего сайта.



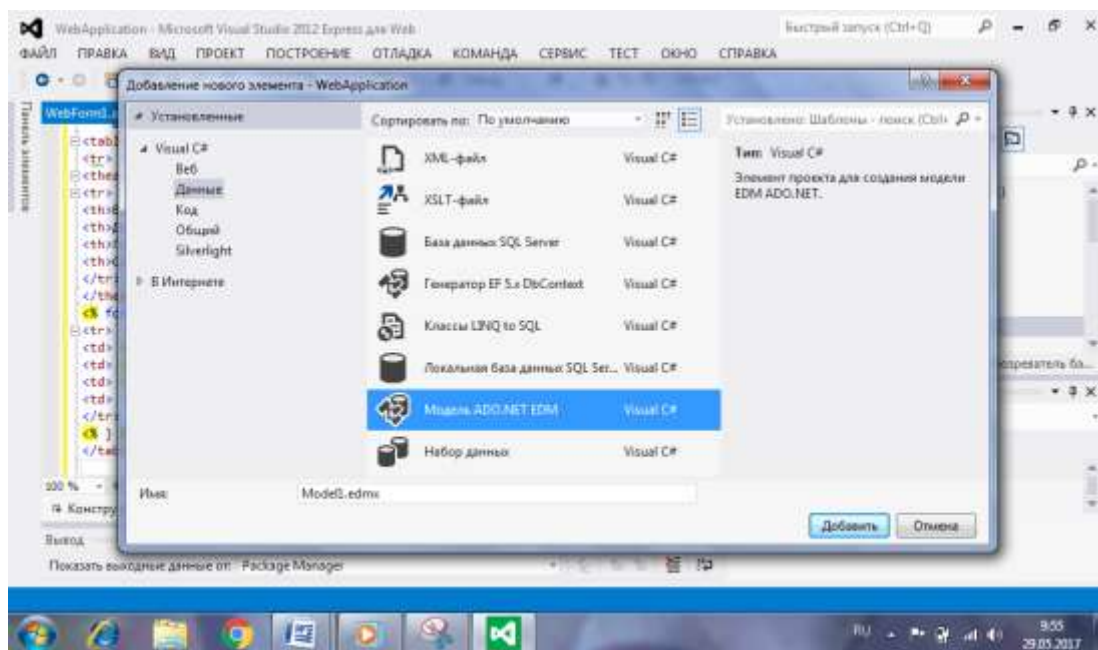


Рисунок 13 – Добавление модели ADO.NET EDM

После, в появившемся окне нажимаем кнопку "Создать соединение", как на рисунке 14. Далее нужно выбрать содержимое модели – "Создать из базы данных", а также выбрать "Имя сервера" и "Имя базы данных" (Рисунок 15).

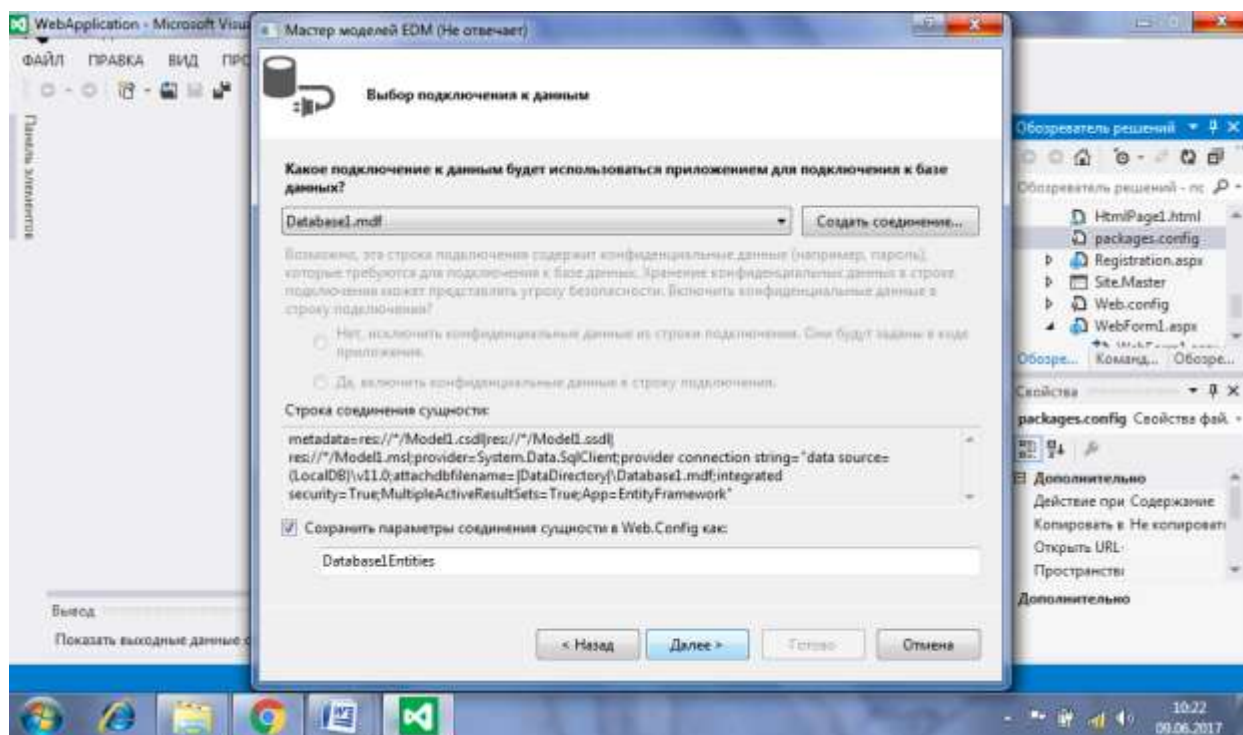


Рисунок 14 – Создать соединение

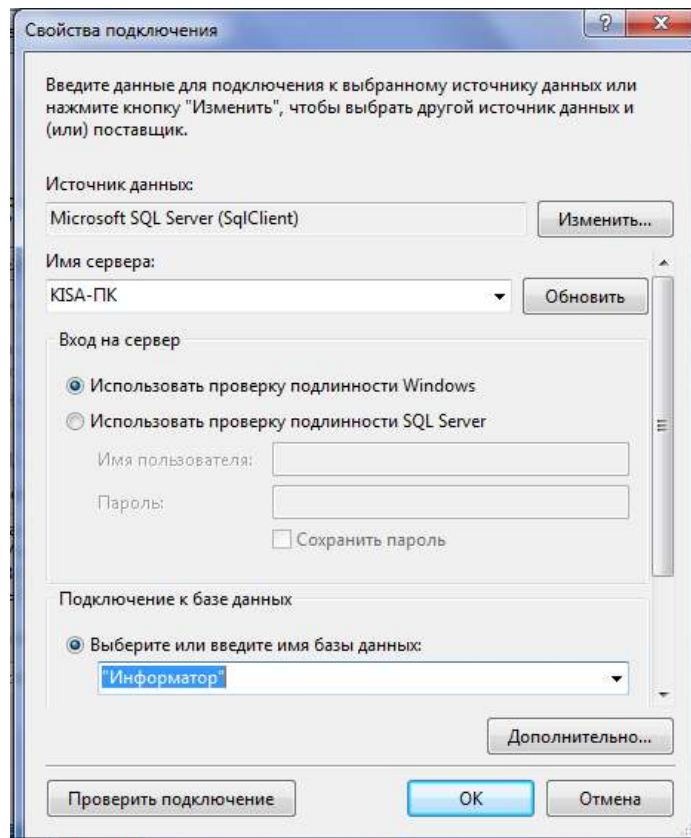


Рисунок 15 – Параметры соединения

После создается соединение и выбираются объекты базы данных, которые нужно включить в модель. В нашем случае это таблицы.

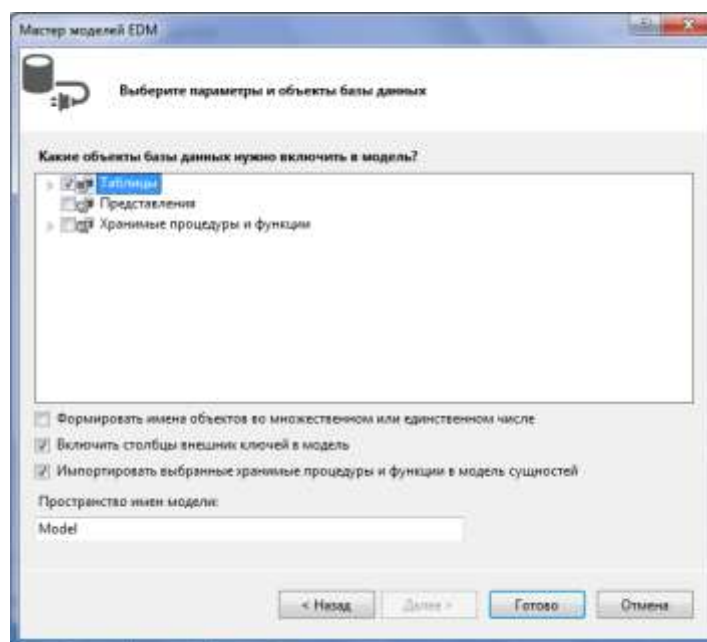


Рисунок 16 – Выбор объектов

Таким образом, создается копия базы данных для дальнейшей работы с веб–приложением (Рисунок 17).

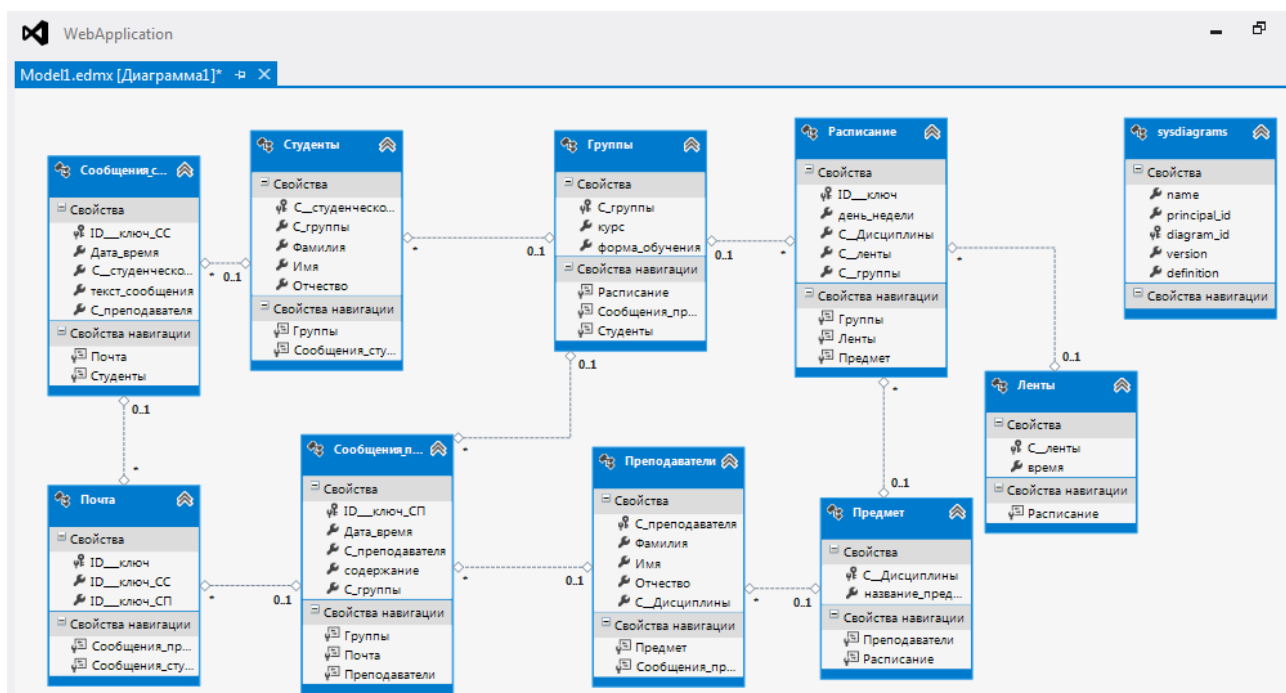


Рисунок 17 – Диаграмма базы данных

### 3.2 Основные элементы создания веб–приложения

Для дальнейшей работы в Visual Studio ознакомимся с его интерфейсом (Рисунок 18).

Основные окна и средства:

- панель инструментов – на панели расположены команды для форматирования текста, поиска текста и т.д. (доступна при работе в конструкторе);
- обозреватель решений – отображает папки и файлы на веб–узле;
- окно документа – отображается активный документ из списка вкладок;
- свойства – позволяет изменять параметры страницы, элементов HTML, элементов управления и т.д.;
- обозреватель серверов и баз данных – отображает подключения;

- вкладки представлений – разные представления одного и того же документа: конструктор (область редактирования), источник – исходный код (HTML–редактор страницы);
- панель элементов – отображены элементы управления и HTML–элементы, которые можно перетаскивать в документ.

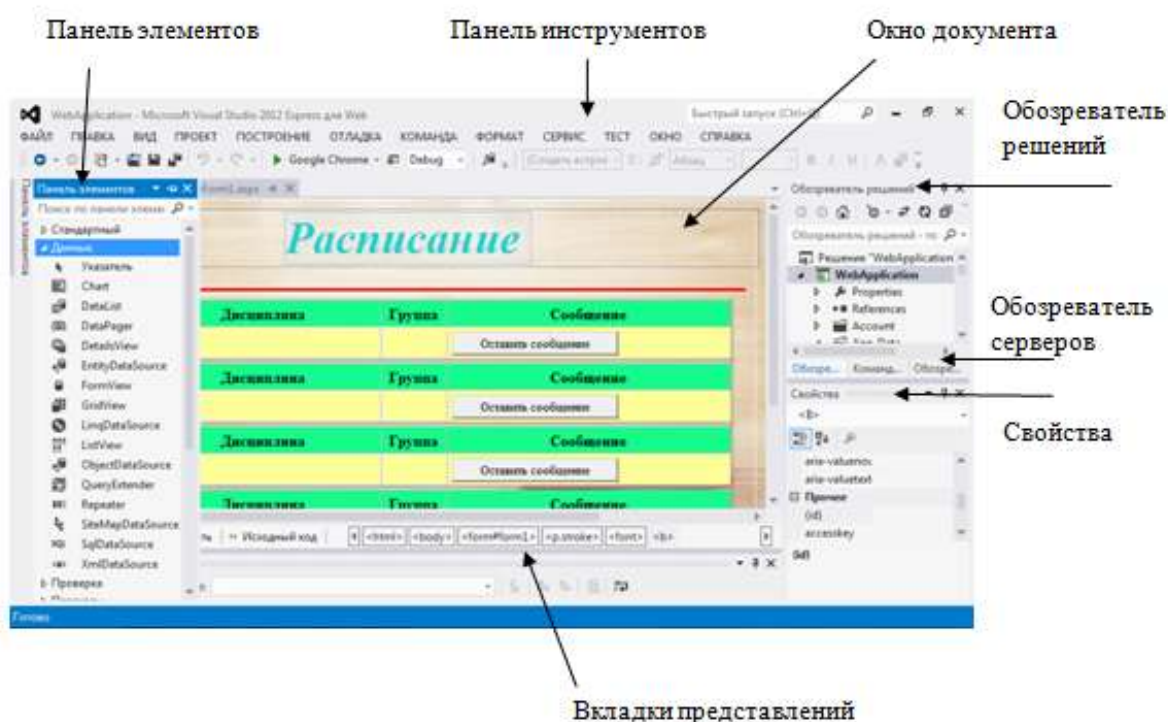


Рисунок 18 – Структура среды Visual Studio

Создавая новый веб–узел Visual Studio добавляет ASP.NET страницу – страницу Web Forms, с именем Default.aspx. Она используется в качестве домашней страницы веб–узла.

Для запуска выполнения страницы в веб–обозревателе можно воспользоваться комбинацией клавиш Ctrl+F5 – в таком случае страница запустится без отладки. Так же можно прибегнуть к меню/отладка/запуск без отладки.

К элементам управления относятся.

- Button – кнопка;
- Label – надпись (метка), не изменяемая пользователем;

- TextBox – текстовое поле;
- ListBox – список;
- CheckBox – флажок;
- RadioButton – переключатель;
- DateTimePicker – позволяет выбрать дату и время из списка (календаря);
- DropDownList – позволяет выбрать элемент из выпадающего списка;
- Panel – панель и т.д.

С помощью обозревателя серверов и баз данных можно просматривать серверные ресурсы и взаимодействовать с ними (Рисунок 19). Можно создавать базы данных и заполнять их, а так же выполнять запросы.

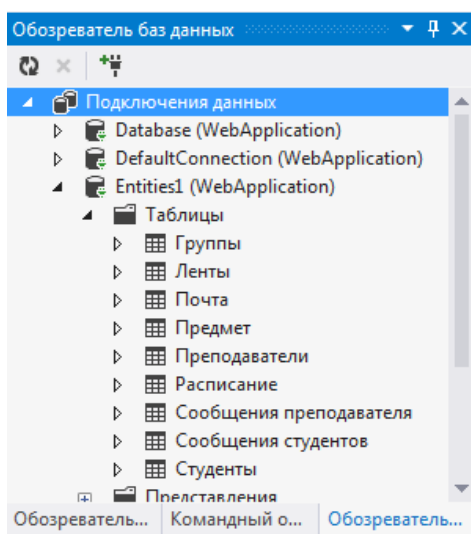


Рисунок 19 – Обозреватель баз данных

### 3.3 Дизайн пользовательского интерфейса

Основная задача интерфейса – обеспечение взаимодействия.

От дизайна пользовательского интерфейса зависит успешность веб-приложения. Важно заинтересовать пользователя и не оттолкнуть чрезмерной навязчивостью, яркостью и пестротой.

Создавая интерфейс приложения необходимо придерживаться следующих аспектов.

Интерфейс должен быть понятным пользователю и удобным для взаимодействия пользователя с приложением. Для понимания приложения уместным будет наличие справки и подсказок. Для удобства желательно применять навигацию и всплывающие списки.

Предоставлять пользователю нужно как можно меньше действий.

Детали интерфейса не должны быть мелкими, их должно быть видно и возможно использовать.

При расположении элементов на странице не стоит забывать о порядке чтения, поэтому более значимые части стоит располагать сверху – вниз, слева – направо.

Кнопки стоит располагать аналогично той операционной системе или программе, в которой привык работать пользователь.

Расположение деталей не должно быть в разных углах страницы, а так же не приветствуется хаотичность расстановки частей приложения.

Соблюдение пропорций и подборка корректной цветовой гаммы может так же отразиться на восприятии пользователя.

### **3.4 Реализация веб–приложения**

Сайт будет состоять из пяти страниц.

Первая страница – авторизация пользователя (Рисунок 20). Здесь пользователь вводит пароль и логин для входа на свою страницу (для каждого преподавателя своя страница, для каждой группы – своя).

При необходимости, возможно, восстановить пароль. Для этого в форме регистрации необходимо указать свой сотовый телефон, на который будет отправлен код восстановления.



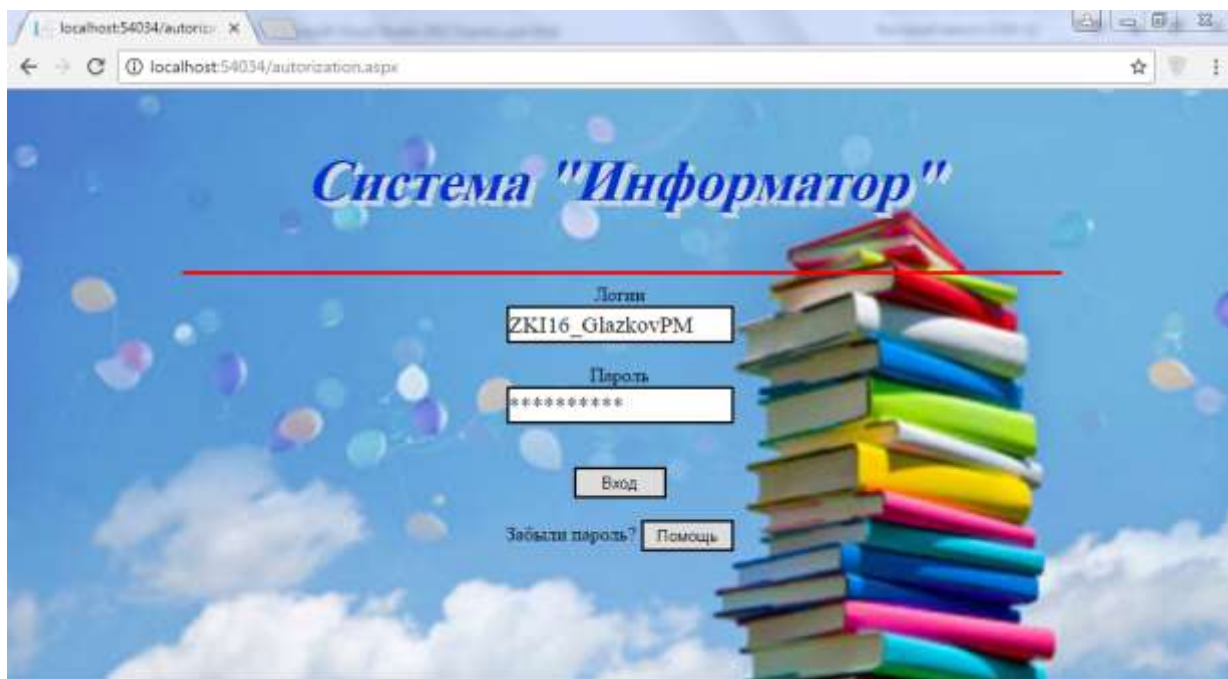


Рисунок 20 – Страница авторизации

Вторая страница – регистрация, если пользователь не зарегистрирован (Рисунок 21). Преподавателей регистрирует администратор, студенты регистрируются сами

Фамилия	Глазков
Имя	Петр
Отчество	Михайлович
№Студенческого билета	120
Группа	ЗКИ16
Сотовый телефон	99452346789
Логин	ZKI12_GlazkovPM
Пароль	1234vujg345
Повторить пароль	1234vujg345

Зарегистрироваться

Рисунок 21 – Страница регистрации

Третья страница – страница преподавателя (Рисунок 22). Так как преподавателей регистрирует администратор, при вводе своих идентификационных данных открывается конкретная страница данного пользователя.

Для осуществления выпадающего списка используется код:

```
<asp:Label ID="Label6" runat="server" Text="Группа" Font-Bold="True" Font-Names="Times New Roman" Font-Size="Large"></asp:Label> /* выводит надпись "Группа" */
<asp:DropDownList ID="CheckBoxList" runat="server"
Font-Names="Times New Roman" Font-Size="Large"> /* создает выпадающий
список и форматирует написанное*/
<asp:ListItem>ЗКИ12</asp:ListItem>
<asp:ListItem>ЗКИ13</asp:ListItem>
<asp:ListItem>ЗКИ16</asp:ListItem>
<asp:ListItem>КИ14</asp:ListItem>
<asp:ListItem>КИ15</asp:ListItem>
<asp:ListItem>КИ16</asp:ListItem></asp:DropDownList> /* конец списка */
<br />
```

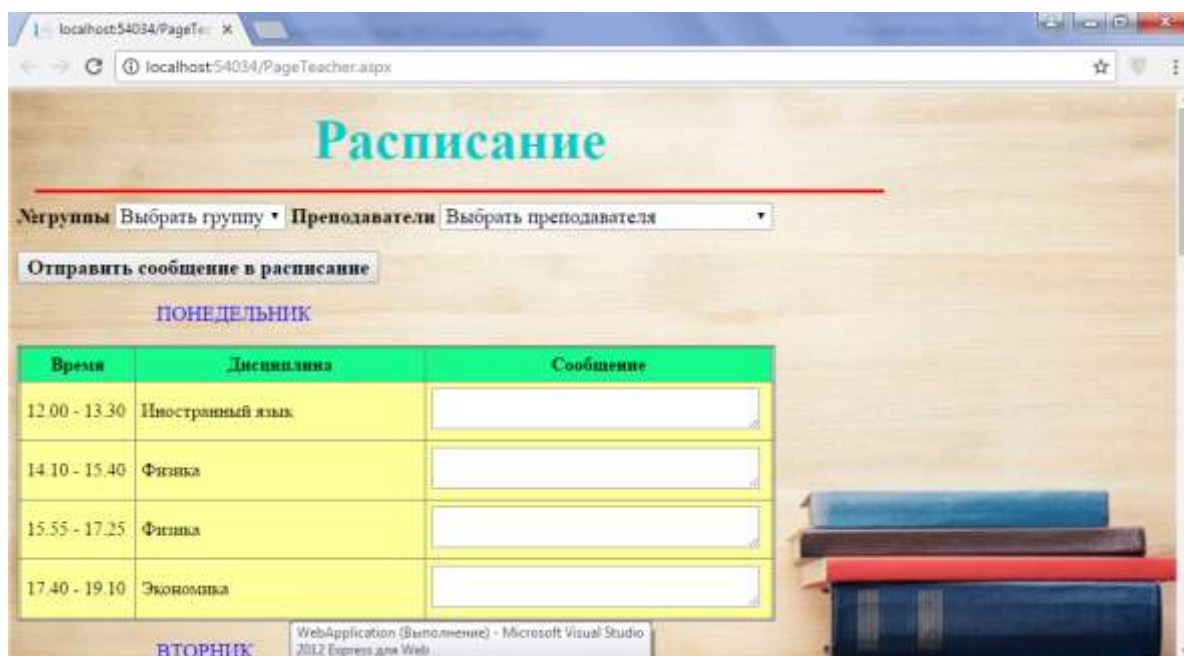


Рисунок 22 – Страница преподавателя

Каждый преподаватель видит свое расписание, при этом он может выбрать расписание конкретной группы и отправить сообщение к



определенной дате и предмету нажав на кнопку "Отправить сообщение в расписание". При этом откроется страница "Сообщение" (Рисунок 24). Так же он имеет доступ к расписанию других преподавателей, но только в режиме чтения.

Четвертая страница – страница студентов и (Рисунок 23). Здесь отображается конкретное расписание той группы, к которой принадлежит студент, и сообщения преподавателей к определенному предмету и дню. В расписании студенты оставлять сообщения не могут, но могут общаться в форуме.

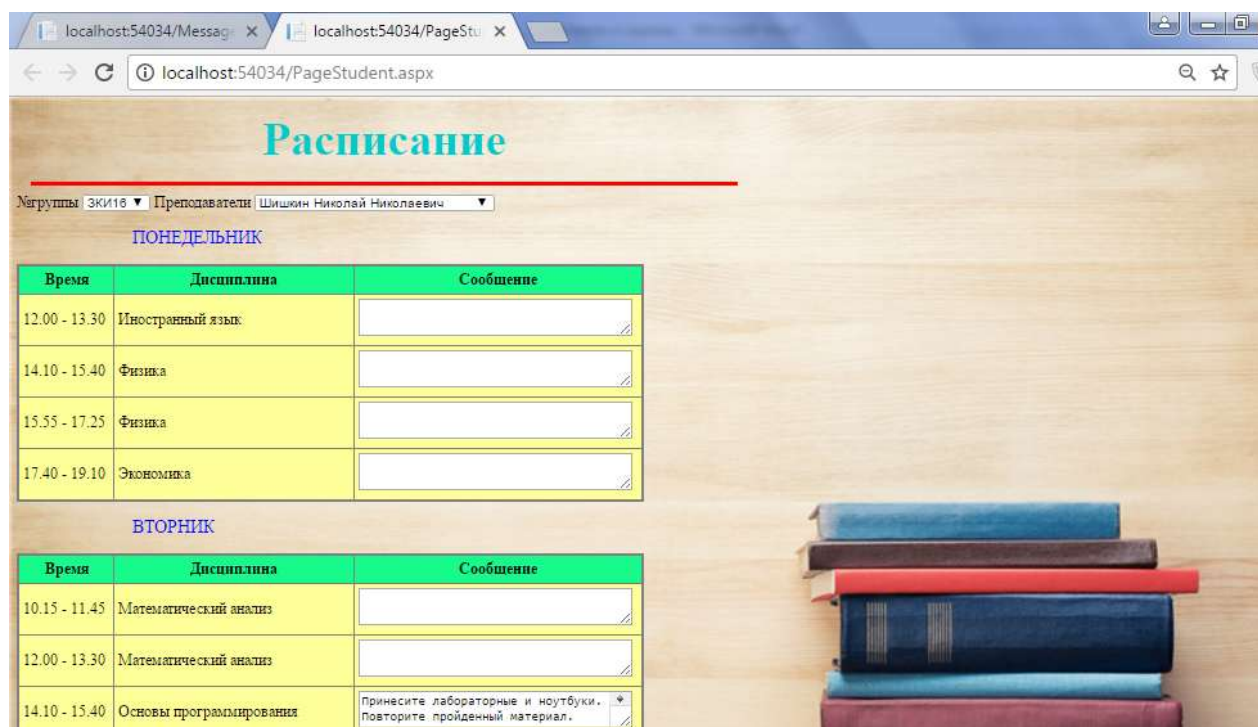


Рисунок 23 – Страница студента

Здесь же находится форум, в котором студенты общаются между собой и с преподавателями (Рисунок 24).

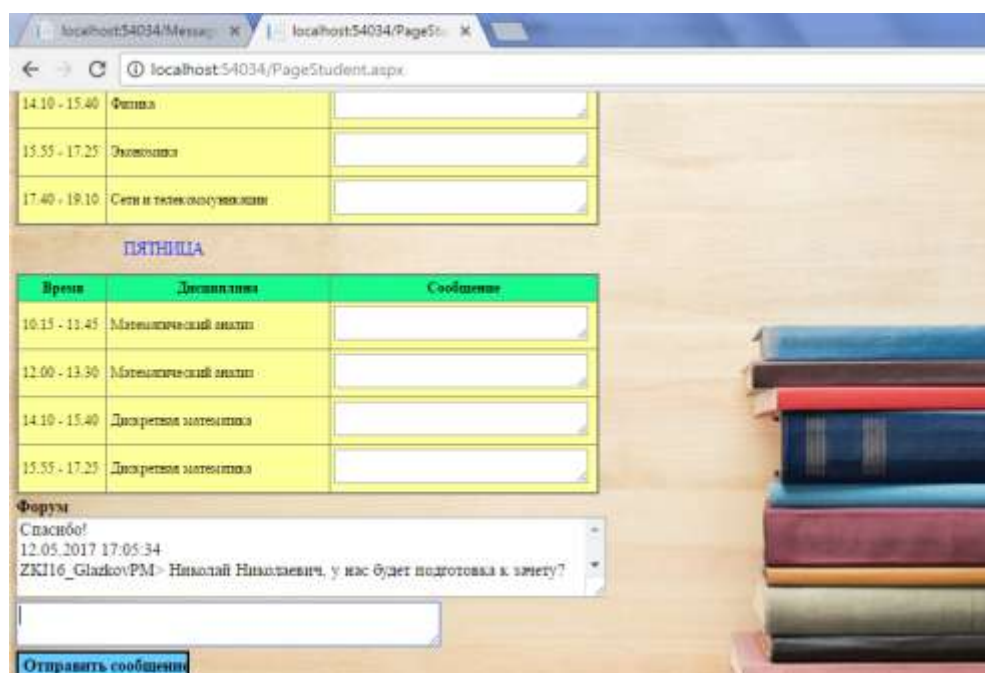


Рисунок 24 – Форум

Пятая страница необходима для написания сообщения для студентов (Рисунок 25). Страница – "Сообщение". Здесь преподаватель выбирает группу, которой необходимо послать сообщение, дату на которую необходимо отобразить сообщение и предмет, к которому относится сообщение.

После отправки сообщения происходит возврат на личную страницу.

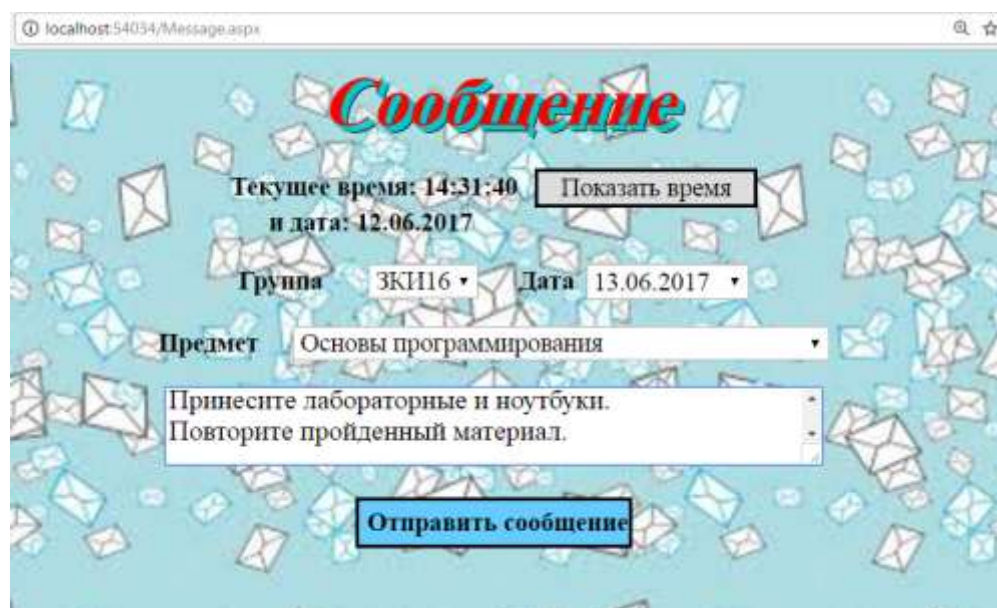


Рисунок 25 – Страница "Сообщение"

Для создания фона используется картинка. Для масштабирования фона предназначено свойство **background-size**, в качестве его значения указывается 100%, тогда фон будет занимать всю ширину окна браузера. Для старых версий браузеров необходимо использовать конкретные свойства с префиксами, как в листинге 1:

Листинг 1

```
body {  
    background: url(images//sky.jpg) 100%; /* задаем фон */  
    -moz-background-size: 100%; /* Firefox 3.6+ */  
    -webkit-background-size: 100%; /* Safari 3.1+ и Chrome 4.0+ */  
    -o-background-size: 100%; /* Opera 9.6+ */  
    background-size: cover; /* современные браузеры */  
}
```

Для создания тени надписи используется листинг 2:

Листинг 2

```
.stroke {  
    font: bold 3em Arial, sans-serif; /* шрифт */  
    color: #180ad4;  
    text-shadow: #00cccc 1px 1px 0, #00cccc 2px 2px 0, /* тень */  
                #00cccc 3px 3px 0, #00cccc 4px 4px 0,  
                #000000 5px 5px 0;  
}
```

Для отображения текущего времени и даты используется листинг 3 и обрабатывается событие нажатия кнопки:

Листинг 3

```
<script runat="server">  
    protected void Button1_Click(object sender, EventArgs e)  
    {  
        Label3.Text = "Текущее время: " + DateTime.Now.ToLongTimeString();  
        Label4.Text = "и дата: " + DateTime.Now.ToString("dd.MM.yyyy");  
    }  
</script>
```

Листинг 4 (событие)

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label3.Text = "Текущее время: " + DateTime.Now.ToLongTimeString();
    Label4.Text = "и дата: " + DateTime.Now.ToString("dd.MM.yyyy");
}
```

Для динамических элементов управления, таких как кнопка, календарь, текстовое поле и т.д., необходимо задавать обработчики событий.

Перечень нескольких основных типов файлов, встречающихся при разработке веб–приложения ASP.NET:

.aspx – веб–страница ASP.NET. Содержит пользовательский интерфейс и базовый код приложения.

.cs – файлы отдельного C# кода. Отделяют приложение от пользовательского интерфейса веб–страницы. Здесь создается обработчик события.

## **ЗАКЛЮЧЕНИЕ**

На практике были применены знания по созданию баз данных с использованием программы MS SQL Management Studio. Рассмотрена технология ASP.NET, позволяющая приложению управлять и взаимодействовать с серверным хранилищем данных. Технология ASP.NET в полной мере способна предоставить механизм для доступа к любому источнику данных, тем самым давая разработчику мощный механизм взаимодействия с базами данных, способный реализовать все потребности, возникающие при проектировании информационной системы.

В результате работы разработана база данных с информацией о студентах, преподавателях, расписанием занятий, и веб–приложение реализующее общение студентов с преподавателями, в виде веб–сайта с размещенным в нем расписанием и форумом.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бейли, Л. Изучаем SQL / Л. Бейли. – Санкт–Петербург: Питер, 2012. – 592 с.
2. Бондарь, А.Г. Microsoft SQL Server 2014 / А.Г. Бондарь. – Санкт–Петербург: БХВ–Петербург, 2015. – 592 с.
3. Евсеева, О. Н. РАБОТА С БАЗАМИ ДАННЫХ НА ЯЗЫКЕ C#. ТЕХНОЛОГИЯ ADO .NET: учебное пособие / О. Н. Евсеева, А. Б. Шамшев. – Ульяновск: УлГТУ, 2012. – 170 с.
4. Климов, А. П. C# Советы программистам / А. П. Климов. – Санкт–Петербург: БХВ–Петербург, 2012. – 544 с.
5. Культин, Н. Б. Microsoft Visual C# в задачах и примерах: 2–е изд. / Н. Б. Культин. – Санкт–Петербург: БХВ–Петербург, 2015. – 320 с.
6. Пахомов, Б. И. C# для начинающих / Б. И. Пахомов. – Санкт–Петербург: БХВ–Петербург, 2014. – 432 с.
7. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен. – Москва: Вильямс, 2013. – 1312 с.
8. Форта, Б. SQL за 10 минут / Б. Форта. – Москва: Вильямс, 2015. – 288 с.
9. "Википедия" [Электронный ресурс] // Справочная информационная система. – Режим доступа: <https://ru.wikipedia.org/wiki/Веб-сервер>.
10. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Пошаговое руководство. Добавление платформы динамических данных на существующие веб-сайты ASP.NET. // Официальный сайт "Майкрософт". – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/ee923692\(v=vs.100\).aspx](https://msdn.microsoft.com/ru-ru/library/ee923692(v=vs.100).aspx).
11. Ресурс для пользователей продукцией Microsoft [Электронный ресурс]: Описание основных приемов нормализации базы данных // Официальный сайт "Майкрософт". – Режим доступа:

<https://support.microsoft.com/ru-ru/help/283878/description-of-the-database-normalization-basics>.

12. "Викиучебник" [Электронный ресурс] // Справочная информационная система. – Режим доступа: [https://ru.wikibooks.org/wiki/Средство\\_разработки\\_веб-приложений\\_Visual\\_Studio\\_на\\_основе\\_ASP.NET/Visual\\_Studio\\_и\\_технология\\_ASP.NET](https://ru.wikibooks.org/wiki/Средство_разработки_веб-приложений_Visual_Studio_на_основе_ASP.NET/Visual_Studio_и_технология_ASP.NET).

13. "Википедия" [Электронный ресурс] // Справочная информационная система. – Режим доступа: [https://ru.wikipedia.org/wiki/ER-модель\\_данных](https://ru.wikipedia.org/wiki/ER-модель_данных).

14. Всё о языках программирования [Электронный ресурс]: С#. // Официальный сайт "АльтКод". – Режим доступа: <http://altcode.ru/c-sh/>.

15. Государство. Бизнес. ИТ [Электронный ресурс]: Microsoft Visual Studio. // Официальный сайт "Тадвизэр". – Режим доступа: [www.tadviser.ru/index.php/Продукт:Microsoft\\_Visual\\_Studio](http://www.tadviser.ru/index.php/Продукт:Microsoft_Visual_Studio).

# ПРИЛОЖЕНИЕ А

(обязательное)

## Сущности и их атрибуты

Таблица А.1 – Сущности и их атрибуты

Преподаватели	Предмет	Расписание	Ленты	Группы	Студенты	Сообщение преподавателя	Сообщения студентов	Почта
№ преподавателя	№ дисциплины	ID – ключ	№ ленты	№ группы	№ студенческого билета	ID –ключ	ID –ключ	ID – ключ
Фамилия	Название предмета	День недели	время	курс	№ группы	Дата	Дата	ID – ключ СС
Имя		№ дисциплины		Форма обучения	Фамилия	№ группы	№ студенческого билета	ID – ключ СП
Отчество		№ ленты			Имя	№ преподавателя	Содержание	
№ дисциплины		№ группы			Отчество	Содержание		



Федеральное государственное автономное образовательное  
учреждение высшего образования  
"СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ"

Институт космических и информационных технологий

институт

Кафедра "Информационные системы"

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

С.А. Виденин

подпись

инициалы, фамилия

"\_\_" \_\_\_\_\_ 2017 г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

**в форме** бакалаврской работы

бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Казиной Елене Олеговне  
фамилия, имя, отчество  
 Группа ЗКИ12-07Б-1 Направление (специальность) 09.03.01  
номер код

Информатика и вычислительная техника  
наименование

Тема выпускной квалификационной работы \_\_\_\_\_  
Разработка системы "Информатор"

Утверждена приказом по университету № 2904/с от 07.03.2017

Руководитель ВКР И.А. Ефремов, доцент кафедры "Информационные системы", к. ф–м. н., ИКИТ СФУ

---

инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР:

- штатное расписание;
- учебный план;
- электронный журнал;
- контингент.

Перечень разделов ВКР: введение, теоретическая часть, база данных, создание веб-приложения "Информатор", заключение.

Перечень графического материала: презентационные слайды PowerPoint.

Руководитель ВКР \_\_\_\_\_ И.А. Ефремов \_\_\_\_\_  
подпись инициалы и фамилия

Задание принял к исполнению \_\_\_\_\_ Е.О. Казина  
подпись инициалы и фамилия студента

" 29 " марта 2017 г.